Enabling Intelligent Platform Management Interface (IPMI) Through Standard Building Blocks

Intel Corporation / OSA Technologies*

Abstract	3
Standards = Savings	3
Standard Hardware Building Blocks	3
Focus on Differentiation	3
A Brief Overview of IPMI	3
What is the Intelligent Platform Management Interface (IPMI)?	3
Overall Advantages of IPMI	3
Applications of IPMI in the Embedded Space	4
IPMI Implementation	4
CompactPCI* Enclosure with IPMI	4
BMC/PMC Selection Guide	4
Complement IPMI and Intel® Architecture	5
Intel Processor Signals	5
Intel® Xeon™ Processor PIROM	7
Intel Chipset Signals	7
Intel Chipset Management Registers	8
Intel Chipset Functions	8
Intel Chipset Manageability Features	8
OSA Firmware and Software Stack	9
OSA IPMI Firmware	9
OSA IPMI Provider Library	10
OSA Repository	10
OSA Connectors	10
OSA Windows Management Instrumentation (WMI) Mapper*	10
Acquiring OSA firmware	10
IPMI-Related Implications of PICMG* 3.0	11
Conclusion	11
References & Resources	11

Abstract

In the embedded world, Telecommunications Equipment Manufacturers (TEMs) and Original Equipment Manufacturers (OEMs) are replacing expensive proprietary product technologies with solutions that are compliant with industry standards. The economics of these industry-standard solutions are compelling.

Intel focuses much of its research and development around standard-based technologies. Intel is currently facilitating the transition from proprietary solutions in the communications infrastructure market to next-generation solutions based on AdvancedTCA* (the PICMG* v3.0 standard). Intel also endorses the Intelligent Platform Management Interface (IPMI) specification as a critical systems management foundation. The synergy between IPMI and other industry standards like PICMG 3.0 adds value beyond simple cost savings. IPMI is the first step toward the next generation of reliable, industry-standard, telecommunications equipment.

To enable the transition to this next generation of equipment, this article enumerates specific steps and technical implementation guidelines that allows customers to use IPMI through Intel Architecture and reap the economic benefits of industry standards. We introduce OSA Technologies* (OSA), as one example of a provider of IPMI firmware/software building blocks. OSA Technologies' firmware/software stack can be used in concert with Intel® Architecture (IA) Reference Designs to deploy and maintain IPMI both quickly and easily.

Standards = Savings

Standard Hardware Building Blocks

Open hardware standards result in high-volume, high-performance, low-cost, interoperable hardware building blocks. These hardware building blocks now provide a foundation for a variety of telecommunications and data communications applications. Telecommunication Equipment Manufacturers (TEMs) and Original Equipment Manufacturers (OEMs) are becoming increasingly encumbered by proprietary hardware. The decrease in available resources across the industry is driving these manufacturers to look toward open standards to deliver products that are fast and cheap.

Focus on Differentiation

By using industry-standard hardware and systems management technologies instead of maintaining proprietary solutions, TEMs and OEMs can increase ROI by focusing on their competitive differentiators. Successful TEMs and OEMs are realizing that they maximize added value by highlighting their differentiators: the features and functions of their equipment. Mature and well-supported industry standards play a significant role in the classic make/buy decision; the time is right to move to AdvancedTCA (PICMG v3.0) and IPMI.

A Brief Overview of IPMI

What is the Intelligent Platform Management Interface (IPMI)?

IPMI is an extensible standard that defines how users can monitor system hardware and sensors, control system components and log important system events. It is an open-standard hardware manageability interface specification. IPMI defines a specific way for embedded management subsystems to communicate with the following other systems:

- Main system processing units (i.e. CPUs)
- Other embedded management subsystems
- Remote management applications (over serial lines, LANs, etc.)

At the heart of IPMI management subsystems are embedded management microcontrollers: BMCs (baseboard management controllers), EMCs (enclosure management controllers), and PMCs (peripheral management controllers).

Overall Advantages of IPMI

Among the advantages of IPMI are:

- IPMI facilitates processor-independent monitoring
- IPMI is a standard that shapes a consistent hardware and software framework
- IPMI implementations reduce development time by handling standard failures

In some systems, the main processor must either constantly poll subsystems or deal directly with common subsystem faults and alerts. With IPMI, this processing burden shifts to cost-effective, IPMI-compliant microcontrollers. The IPMI microcontroller handles system event management, freeing the main processor for other tasks and acting as an independent entity in cases where the main processor is failing.

Rather than deal with hardware event management tasks using proprietary subsystems or load the main processor with unpredictable hardware event handling, it is possible to use a standard IPMI compliant sub-system. The standard allows for more reuse of hardware and software elements.

By taking advantage of IPMI, TEMs and OEMs have an opportunity to handle standard failures in a uniform way. OSA Technologies is one example of a supplier of IPMI firmware that includes standard building blocks to handle normal failure conditions.

Applications of IPMI in the Embedded Space

In some cases, telecommunications customers place equipment in remote locations where Ethernet access is not available. For example, consider a remote DSLAM where trunk lines are entering the device and phone lines are leaving the device. If the DSLAM experiences a failure situation, the system can alert the appropriate personnel via a modem link. Furthermore, these personnel could use the same modem link to initiate diagnostic routines or cause a device reset.

IPMI Implementation

CompactPCI* Enclosure with IPMI

The following high-level diagram shows how the various parts of an enclosure relate to one another in the IPMI context. The diagram includes some sample cPCI cards that are used by TEMs and OEMs.

BMC/PMC Selection Guide

Hardware designers need to take a few selection criteria into account when deciding which BMC to purchase. The following is a non-exhaustive discussion of some items to consider when selecting a BMC that will work best with Intel Architecture.

System Interfaces

Hardware designers should choose a BMC with multiple built-in system interfaces. KCS (Keyboard Controller Style) is popular, BT (Block Transfer) is fast, and SMIC (System Management Interface) facilitates a connection to an ASIC or FPGA.

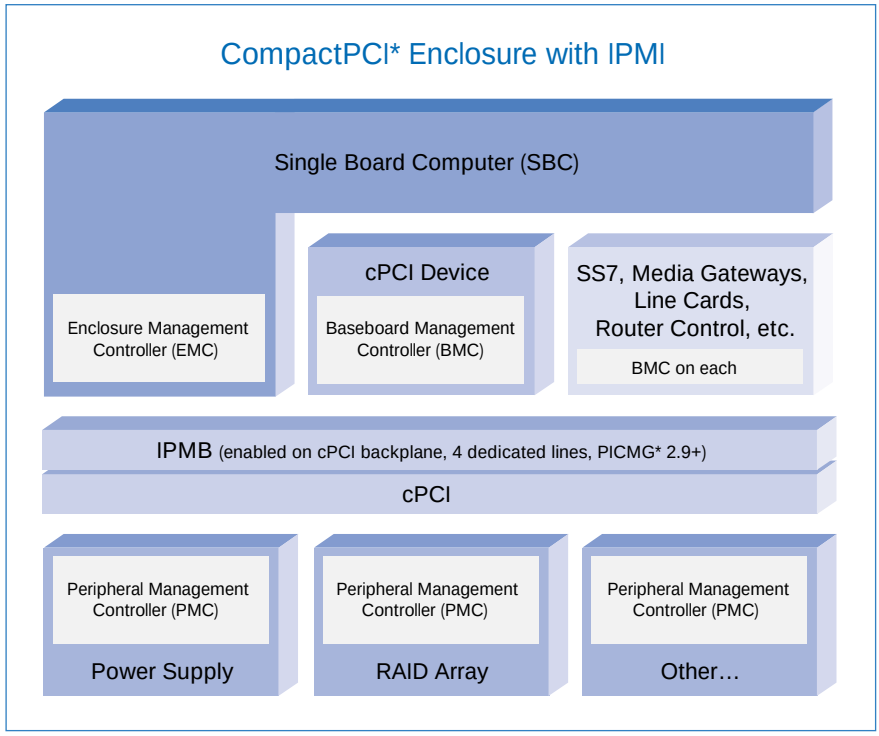

Figure 1. Diagram of cPCI Enclosure With IPMI

A BMC-to-chipset interface with minimal connections is preferable. The new Intel Architecture chipsets are most amenable to LPC (Low Pin Count) connections. The BMC should also provide system interface interrupt support.

I²C/SMBus Support

The BMC should support at least three master-slave interfaces: a 5V interface for IPMB, a 3.3V interface for the PCI management bus and an additional interface for LAN access or expansion. Each of the interfaces should support multiple slave addresses.

Designers should also find a BMC that supports multiple private management busses. This reduces the need for external I²C multiplexing. The timing on the BMC should be compatible with both SMBus and I²C standards.

UARTs and Timers

Internal UARTs in the BMC should support hardware handshaking. The BMC should include a built-in UART for direct ICMB support and an additional UART for IPMI-over-modem applications. A firmware timer should be included for firmware scheduling.

Interrupt Support

Hardware designers should find a BMC that supports multiple internal and external interrupts as well as multi-level interrupts. This facilitates asynchronous event capture, efficient code performance and deterministic prioritization.

Fan Monitoring and Control

In order to monitor fan speed, the BMC should include timers or counters. The counters accumulate counts per unit time to measure speed. A pulse-width modulator or D/A on the BMC can control fan speed.

Built-in Analog-to-Digital

A/D converters on the BMC convert on-board voltage levels such as 3.3V, 5V, 12V, etc. to digital representations. Frequently designers need to accommodate the measurement of at least seven voltages with these A/D converters. At least 8-bit accuracy, +/- 1-bit tolerance, and +/- 1% accuracy are recommended.

Extensibility and Scalability

Designers should make sure that the BMC includes enough ROM/RAM scalability to facilitate additional features, sufficient performance headroom and external expansion capability.

Software Support

BMC vendors offer IPMI firmware for their BMC and/or Enclosure Management Controllers. OSA Technologies, discussed later in this paper, offers standard and custom firmware and software solutions for BMCs.

Complement IPMI with Intel® Architecture

Intel and Hewlett Packard* jointly developed IPMI to help drive down the recurring systems administration component of the "total cost of ownership" (TCO) equation. To take full advantage of IPMI, Intel designs specific features into processors and chipsets in order to meet demanding manageability requirements from a wide range of customers. These customers know how important reliability, serviceability, availability and fault tolerance are in the embedded space. The following is a discussion of features specific to Intel Architecture processors and chipsets that complement IPMI-compliant management subsystems.

The following diagram demonstrates how an Intel Architecture processor and chipset interact with a BMC/PMC in a cPCI design. In the next few sections, we discuss the specific Intel Architecture CPU and chipset status and control lines shown in this diagram, such as THERMTRIP# and SERR#. We also discuss how IPMI firmware might take advantage of these lines, using OSA Technologies' firmware as an example.

Intel Processor Signals

Some of the IPMI-related signals designed into IA processors include:

- PROCHOT# (Maximum safe operating temperature reached)
- THERMTRIP# (Processor junction temperature critical)
- VID, 5 pins (For selection of power supply voltages)
- IERR# (Internal processor error)

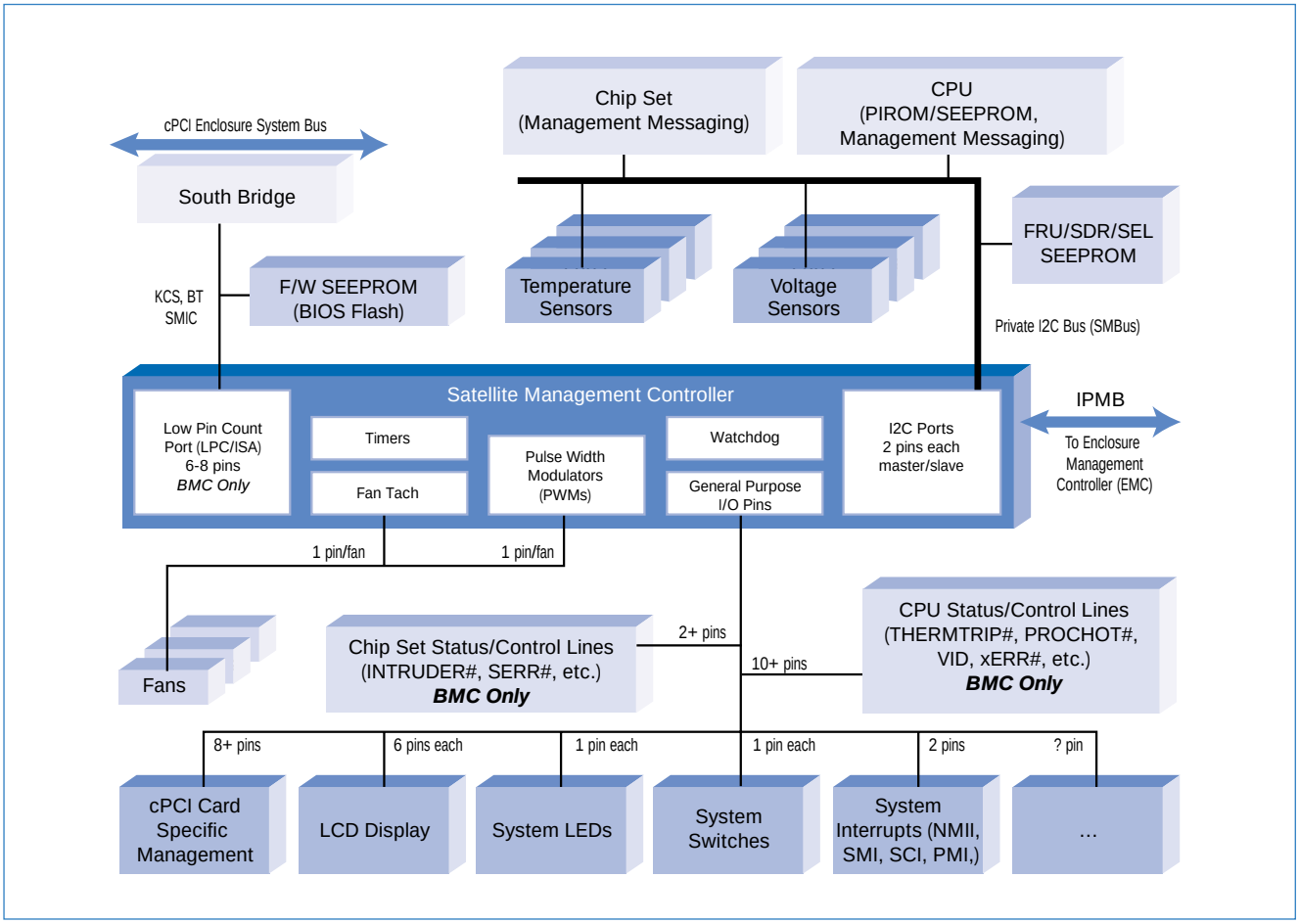

Figure 2. Example cPCI Satellite Management Controller Block Diagram

- FERR# (Floating point error from numeric coprocessor)
- AERR# (Address error)
- BERR# (Bus error)

PROCHOT# and THERMTRIP#

Processors assert PROCHOT# when activating a thermal control circuit that can automatically throttle processor clock speed. When the junction temperature exceeds a critical threshold, the processor asserts THERMTRIP#. Upon a THERMTRIP# trigger, Intel processors attempt to stop internal clocks and halt program execution to reduce internal temperature and avoid processor damage.

Through IPMI, OSA firmware delivers even more options. For many Intel Architecture systems, sophisticated OSA firmware can throttle the processor clock, increase fan speeds or raise system interrupts tied to OSA handlers for advanced configuration and power interface (ACPI) functions. OSA firmware logs such processor thermal alarms/warnings to the IPMI defined system event log (SEL). If there are too many thermal events within a configurable time frame, the firmware generates an IPMI defined platform event trap (PET). This reliable form of an SNMP trap can be routed to any number of management applications (from OSA and others).

VID

Five VID signals on the processor may be used to control the voltage of the VRM (voltage regulator module) and the voltage regulator. When these VID signals are wired to the GPIO pins of the BMC, OSA firmware can detect early signs of VRM failure. Each VRM has a defined set of operational tolerances that OSA firmware can continuously monitor via the five VID bits. Deviations from these tolerances, if they occur more frequently than some configurable threshold value, can trigger one of the following IPMI events: a platform event trap (PET), a system event log (SEL) entry, or even a pager transaction. By proactively monitoring the VRM, IPMI allows next-generation telecom equipment to call before it fails.

In particularly dire situations, OSA firmware can take direct control of the VRM to temporarily counter VRM skew. This is possible only if the processor's VRM control lines are wired to the BMC. While the BMC is countering skew, OSA firmware logic can trigger system interrupts to initiate an orderly system shutdown. The optimal configuration of firmware logic will depend upon the application.

IERR# and FERR#

The IERR# and FERR# pins are open-drain output signals. IERR# indicates a catastrophic error and is pulled up when the processor is functioning normally. FERR# designates that a floating-point error has occurred. Since many next generation systems use multiple processors, OSA firmware can monitor these lines to determine which processor has faulted, log the failure in the system event log (SEL), generate platform event traps (PET), initiate paging notifications, etc. By wiring interrupt lines to each of the processors in your system, OSA firmware can alert the operating systems running on working processors of the failed processor's condition. The working processors can then reconfigure their operating systems to work around the failure and minimize service disruption.

AERR#

AERR# is driven by system bus agents to indicate that one of the agents has observed an address bus parity error. If used, AERR# must connect the appropriate pins on processor system bus agents. An IPMI-compliant BMC might be one of these system bus agents.

OSA firmware facilitates the standard methods of logging AERR# detections in the system event log (SEL) and raising alerts such as platform event traps and pager notifications. Furthermore, by exposing unfavorable trends in AERR# events, OSA firmware enables predictive fault detection.

BERR#

BERR# indicates that an unrecoverable bus error occurred without a bus protocol violation. Processor system bus agents may assert BERR#. Not all Intel processors observe BERR# assertions, and thus a BMC may be used to log and respond to BERR# assertions. In fact, analyzing the frequency of BERR# occurrences is a good way to predict hardware failures. OSA's componentized firmware architecture makes it easy to add/upgrade to firmware that might later monitor and process signals like BERR#. We recommend that CPU signal lines related to manageability connect to the BMC. Even though the BMC may not use these signals initially, putting the connections in place provides for an upgrade path.

Intel® Xeon™ Processor PIROM

The Intel® Xeon™ Processor includes a write-protected PIROM (Processor Information ROM), which contains information that the BMC can use to enhance management capabilities. For example, the PIROM contains values such as the processor VID, the minimum core voltage and the maximum case temperature. Since the BMC is controlling and monitoring these items, BMC firmware can make more intelligent decisions based on the information in the PIROM. The BMC reads the information in the PIROM via the SMBus.

Intel Chipset Signals

The following are some of the signals on IA chipsets relating to IPMI:

- INTRUDER# (Switch input to ICH)
- SERR# (System error input to chipset)

INTRUDER#

The chipset can monitor INTRUDER#, a switch input to the ICH, to determine whether the chassis has been opened. The BMC can read the intruder detect bit register in the SMBus slave interface to log chassis tamper scenarios. The BMC may also monitor INTRUDER# directly, allowing OSA firmware to log intrusion detection events in the system event log (SEL), raise platform event traps (PETs), page security personnel, etc.

SERR#

IA chipsets can monitor SERR# (system errors) and raise interrupts to alert the system. Used in conjunction with the chipset, a BMC along with OSA firmware can further augment manageability by logging errors in the SEL (system event log). Recording SERR# occurrences supports proactive failure prediction.

Intel Chipset Management Registers

Intel Architecture chipsets also include management-related registers accessible by the BMC via the SMBus slave interface such as:

- Intruder detect bit
- Boot status
- Unprogrammed FWH bit
- System Power State
- Frequency Strap Register

Intruder Detect Bit

In cases where there exists a shortage of GPIO lines on the BMC, the BMC can allow the ICH to monitor intrusion detections and then read the intruder detect bit over the SMBus.

Boot Status and Unprogrammed FWH Bits

The boot status bit is set when a boot failure occurs. If the Unprogrammed FWH bit is set, then the FWH is probably blank.

Intel Chipset Functions

The BMC can command IA chipsets to perform certain functions:

- WAKE/SMI# (Wakes system if not awake, otherwise generates an SMI)
- Unconditional powerdown (Same effect as Power Button override)
- Hard reset without cycling (Does not cycle the power supply)
- Hard reset system (Cycles the power supply)
- Disable the TCO messages (Suppresses heartbeat messages)
- WDRELOAD (Reloads the watchdog timer)

Since the BMC has control over these functions, a user could initiate these commands from a remote location.

Intel Chipset Manageability Features

IA chipsets include additional manageability features such as:

- Processor present detection
- Various error detection capabilities (ex. an ECC error can trigger an interrupt)

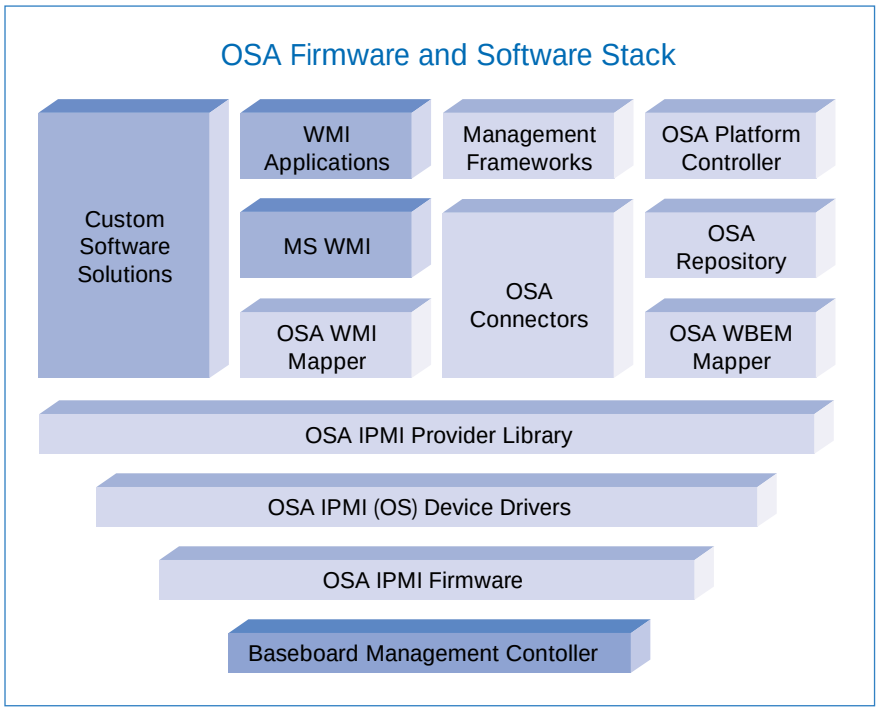

Figure 3. Example Firmware/Software Stack

- Bad firmware detect
- Hiding PCI devices
- Voltage ID reads via GPIO

After selecting a BMC and a hardware platform, designers must decide how to add IPMI firmware and software to their systems. Designers have many options at this point, including developing their own firmware or utilizing a BMC vendor's firmware. Another possible solution would be to commission the help of a company such as OSA Technologies, a provider of IPMI firmware/software building blocks.

OSA Firmware and Software Stack

OSA IPMI Firmware

While OSA firmware provides a rich set of standard facilities, starting with IPMI v1.5 features and functions, OSA firmware architecture supports telecommunications equipment specific firmware "modules". Many customers benefit by taking specific advantages of specialized equipment features. The specialized manageability features in Intel Architecture are a good example. Customers can also contract with OSA to provide proprietary firmware features.

To satisfy the proprietary needs of specific telecommunications equipment manufacturers, OSA customizes each delivered firmware stack. For example, OSA's custom solutions can take advantage of the hardware features specific to Intel Architecture. The following are some of the features included in OSA IPMI firmware:

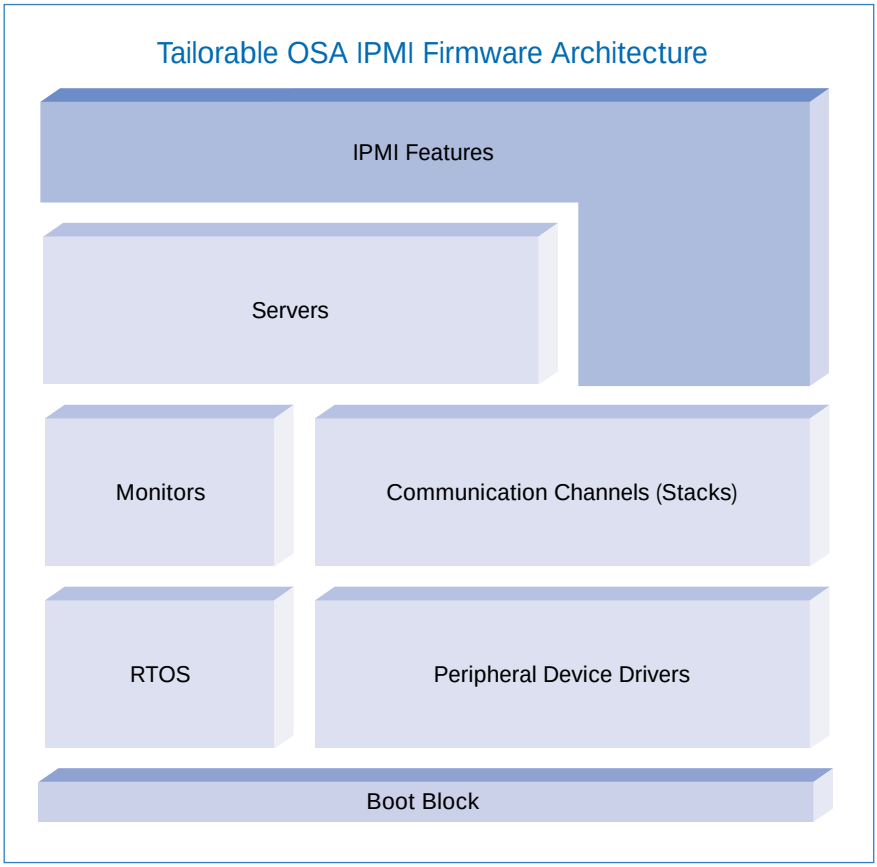

Figure 4. Tailorable OSA IPMI Firmware Architecture

- Comprehensive IPMI v1.5 support, continually maintained for latest errata/addenda
- Flexible configuration for a wide variety of BMC environments
- Modular component architecture for memory-constrained sub-setting
- System development kit (SDK) for proprietary extensions
- Peripheral driver tool box supports Heceta* concentrators and a large number of sensors, power supplies, etc.
- Communication channels such as I^2C, IPMB, ICMB, RS232, RS485, Emergency Management Protocol (EMP), RMCP/ASF tunneling to/from shared NIC side-band, IP, TCP, UDP (dedicated NIC)
- Advanced features such as secure/remote upgrade

The modular nature of the OSA firmware architecture accommodates many degrees of functionality. Additionally, OSA's firmware architecture uses configuration data files to quickly reconfigure general-purpose input/output (GPIO) pin

assignments. This makes it quick and easy to expand OSA's standard microcontroller firmware stack for specific telecommunications equipment designs.

OSA IPMI Provider Library

The OSA IPMI Provider Library yields a convenient, native (compiled C code) interface on platforms ranging from real-time operating systems (RTOSs) to Microsoft Windows* and Unix*/Linux*. A variety of bindings are available for other programming languages like Java. The OSA IPMI Provider Library simplifies almost every aspect of IPMI support:

- Comprehensive, proven and optimized IPMI v1.5
- Automated IPMI session management
- Calls to multiple sensors at the same time
- Consolidated calls for user information/management
- Robust handling of BMC data transfer limits

The IPMI Provider Library communicates with an OSA BMC device driver that is resident in the OS. The device driver communicates with the BMC through one of the three IPMI defined system interfaces (KBC, BT and/or SMIC). Customers can use the IPMI Provider Library in conjunction with this device driver to speed up IPMI software development projects.

OSA Repository

The OSA repository collects management-related data via IPMI and other sources. This management history assists in proactively predicting failures and capitalizing on many other advanced management functions. For example, the management history facilitates automation of various aspects of sizing and capacity planning. Translator components make management repository data available via protocols like TL/1, CMIS/CMIP, SNMP and WBEM/CIM. Contact OSA about translators for Telcordia NMA*, Telcordia OPS/INE* and the necessary OSMINE certification.

OSA's WBEM Mapper supports the OSA repository (and/or other WBEM/CIM compliant common information model object managers – CIMOMs).

OSA Connectors

OSA connectors communicate with popular server management application frameworks. Connectors make IPMI features and functions available to frameworks like Hewlett Packard's OpenView* and Computer Associates Unicenter TNG*. Contact OSA about connectors for telecommunications management frameworks.

OSA Windows Management Instrumentation (WMI) Mapper*

The OSA WMI Mapper simplifies access to IPMI features and functions available through Microsoft's WMI*. The WMI Mapper reduces time-to-market when customers need to support Windows-based appliances or Windows-based management applications.

Acquiring OSA firmware

OSA works with many management microcontroller vendors to maintain OSA's base IPMI v1.5 firmware stack for each management controller. The base firmware stack is largely C code that is cross-compiled for a variety of microcontrollers. OSA supports management controllers ranging from sophisticated enclosure management controllers (EMCs) to simple peripheral management controllers (PMCs).

As a rule of thumb, allow one month from contract close to delivery from OSA. Purchasing binary images, system development kits or even source distributions should not affect lead times.

In general, OSA expects firmware customization for a specific piece of equipment to take approximately three months from contract close. The actual amount of time depends upon the degree of customization necessary. It is advisable to engage OSA's IPMI design consultants during the design phase to optimize the optimal microcontroller selection and configuration process.

For most firmware deployments, OSA does most (or all) firmware work in parallel with other phases of product development. Engaging with OSA's IPMI design consultants early is the key to minimizing development time.

IPMI-Related Implications of PICMG* 3.0

AdvancedTCA, or PICMG 3.0, draft standards already call for IPMI v.1.5 compliant management controllers. Industry standard hardware management technology was a prominent feature in the PICMG 2.9 standard. With respect to IPMI, designers can expect a smooth transition from cPCI to AdvancedTCA.

One example of how IPMI complements the AdvancedTCA standard is related to thermal stress. ATCA is based on intensive thermal modeling: a wider board pitch and airflow cooling specifications attest to AdvancedTCA's focus on thermal stress. By making intelligent decisions based on signals such as THERMTRIP# and PROCHOT#, Intel Architecture and OSA firmware can deliver a comprehensive thermal manageability solution for next generation telecommunications equipment.

Conclusion

By taking advantage of open standards such as IPMI and AdvancedTCA (PICMG v3.x), telecommunications equipment manufacturers (TEMs) and original equipment manufacturers (OEMs) can realize significant cost savings. In addition, they can increase their ROI by focusing on their competitive differentiators. Intel Architecture processors and chipsets facilitate this increased ROI since they were designed to complement IPMI and AdvancedTCA. OSA Technologies provides extensible, off-the-shelf IPMI firmware and software supporting multiple management controllers. By using Intel Architecture processors and chipsets along with OSA firmware to take full advantage of new industry standard specifications, TEMs and OEMs can reap significant benefits beyond simple cost savings.

References & Resources

[1] Hewlett Packard*: "IPMI: Intelligent Platform Management Interface White Paper", February 1998.

[2] Intel Corporation: "Intel® 82801CA I/O Controller Hub 3-S (ICH3-S) Datasheet" p.154, 206-208.

[3] Intel Corporation: "Intel® Xeon™ Processor with 512KB L2 Cache at 1.80GHz to 2.40GHz Data Sheet", p.78-87, 98-101.

[4] Intel Corporation: "IPMI v. 1.5 Revision 1.1, 2/20/02"

[5] Slaight, Tom. "Server Management Controllers, Sensors, and Tools". Intel Developer Forum, Spring 2001.

[6] http://www.picmg.com/pdf/AdvancedTCAOverview.pdf

0502/LM/QI

251337-001